Buzut

Rechercher...



21 janvier 2017 (update)

101 commandes indispensables sous linux

Linux 52 commentaires

Voici un article qui sera complété au fur et à mesure de mes découvertes. Il condense un peu plus d'une centaine de commandes qu'il est utile de connaitre sous linux, que ce soit sur un desktop/laptop ou un serveur, gardez-les sous la main, elles vous seront toujours utiles ! Gardez bien à l'esprit que ce post ne peut en aucune façon se substituer au fameux « man », c'est plutôt un cookbook des commandes qui reviennent le plus.

Plan de l'article

Commandes spécifiques aux ordis Apple

Commandes de bases, navigation dans les fichiers

`ls`

liste les fichiers d'un dossier. Options : `-a` pour les fichiers cachés, `-l` pour la liste détaillées, `-h` pour les tailles en unités « human readable ».

`cd`

change directory, la commande permet de naviguer dans l'arborescence. Par exemple `cd /var/log` va dans le dossier des logs, quelque soit l'endroit où l'on se trouve puisqu'on a mis le slash de début, lequel indique qu'il s'agit d'une adresse absolue. En revanche, `cd mondossier/images` va dans le répertoire `images` de `mondossier` lequel se trouve à l'endroit où on se situe déjà. Comme on ne met pas de slash de début, il s'agit d'une adresse relative, on ajoute donc ce chemin à celui dans lequel on se trouve déjà.

`du`

disk usage, précise l'espace disque que prend chaque fichier ou dossier (l'option `-h` permet d'obtenir les tailles en « human readable »), tandis que l'option `-c`, très utile également, permet d'obtenir le total des éléments analysés. Cette commande s'avère particulièrement pratique quand `ls -l` ne nous donne pas la taille d'un dossier.

`pwd`

print working directory. Cette commande affiche tout simplement le chemin absolu du dossier dans lequel on se trouve,

```
Buzut$ pwd
/etc/apache2
```

`ssh`

secure shell. Permet de se connecter au shell d'un ordinateur distant et d'y exécuter des commandes.

```
ssh login@ip ou nom_hôte
```

Et voilà tout, pour se déconnecter, ce sera exit, et toutes les commandes habituellement utilisables dans un terminal le sont aussi via ssh. Une option bien pratique de ssh est le tunneling qui permet par exemple de déjouer les pare-feux par la mise en place d'un proxy socks. Si certains ports sont bloqués et vous empêchent de vous servir de tel ou tel service ou d'accéder à tel ou tel site (en vacances en Chine ?), la solution est donc de tout rediriger vers un port local et de laisser la machine distante – qui n'est pas derrière un part feu – accéder aux ressources non autorisées. Nous allons donc rediriger tout notre trafic vers un port prédéfini (en l'occurrence 2013), à travers notre connexion SSH.

```
ssh -D 2013 login@ip_serveur_distant
```

Dernière étape pour que cela fonctionne, il faut que votre trafic sortant soit redirigé vers le port 2013. Vous pouvez le configurer dans votre navigateur et pour l'ensemble des connexions de votre ordinateur dans les préférences réseau.

`clear`

nettoie votre fenêtre de terminal en reléguant tout le texte au dessus (donc accessible avec un scroll) et vous laissant donc face à une fenêtre clean. Bien utile de temps à autre pour y voir plus clair. Le raccourci clavier `ctrl` + `l` fait la même chose.

`ctrl` + `r`

très très pratique, permet de faire une recherche dans l'historique des commandes. Habituellement, vous remontez dans les commandes déjà tapées avec la flèche du haut, eh bien avec `ctrl` + `r`, vous pouvez effectuer une recherche dans cet historique, faites `ctrl` + `r`, puis tapez un bout de la commande que vous voulez rechercher, magique !

`for`

C'est certainement la commande la plus complexe ce cette section, surtout si vous ne programmez pas. `for` est une instruction de boucle. Une boucle permet d'exécuter une action plusieurs fois, sur tous les éléments d'une variable. Par exemple, nous pouvons ainsi très facilement renomer tous les fichiers d'un répertoire pour

remplacer les espaces par des traits d'union.

```
# pour tous les éléments dans le répertoire courant
# l'instruction in place les éléments correspondants dans la varial
for oldname in *
# do exécute l'action en boucle
do
  # on utilise ici sed pour renommer,
  # l'usage de la commande est expliqué plus bas dans l'article
  newname=`echo $oldname | sed -e 's/ /_/g'`

  # on invoque la commande mv pour remplacer l'ancien nom par le no
  mv "$oldname" "$newname"
done
```

Opération sur les fichiers

`cat`
: lire le contenu d'un fichier texte `cat monfichier.txt`

`less`
: fonctionnement similaire à `cat` mais affiche le fichier page par page. C'est donc plus pratique pour les longs fichiers.

`head`
: affiche l'en-tête d'un fichier, l'option `-n` permet de spécifier le nombre de lignes à afficher.

`tail`
: semblable à `head` mais concerne la « queue » du fichier, en d'autres termes, cette commande n'affiche que la fin. Une option très appréciable `-f` pour follow, permet de mettre à jour en temps réel l'affichage de la fin du fichier, ce qui est fort pratique pour suivre l'évolution d'un fichier de logs par exemple ;)

`touch`
: créer un fichier. Il suffit de faire `touch nom_du_fichier`. Comme Franckito le précise dans les commentaires `touch` a pour but premier de modifier l'horodatage d'un fichier. Si vous faites `touch` sur un fichier qui existe déjà, cela actualisera ses dates de dernier accès et modification.

`mkdir`
: créer un dossier, le fonctionnement est le même que celui de la commande `touch`. `mkdir nom_du_dossier`

`cp`
: copy, faire une copie d'un fichier. L'option `-R` permet de réaliser des copies de dossiers entiers.

```
cp fichier_original copie_du_fichier

# on peut aussi placer la copie directement dans un autre dossier
cp fichier_original nom_du_dossier/copie_du_fichier
```

`mv`
: move, permet de déplacer des fichiers/dossiers. La commande mv s'utilise exactement de la même manière que la commande `cp`. En outre, cette commande permet aussi de renommer les fichiers et dossiers en changeant simplement leur nom.

```
mv mon_fichier mon_fichier_new_name
```

`rm`
: remove, supprime des fichiers. `rm nom_du_fichier`. L'option `-f` force la suppression, l'option `-i` demande une confirmation avant suppression, enfin l'option `-r` permet la suppression des dossiers.

`rmdir`
: remove directory, supprime un dossier seulement s'il est vide.

`ln`
: link, créer un lien entre deux fichiers. L'option `-s` permet de créer un lien symbolique.

```
#créer un lien en dur
ln fichier1 fichier2

#créer un lien symbolique
ln -s fichier1 lien_vers_fichier1
```

`wc`

word count, permet de compter le nombre de lignes, de mots et de caractères dans un fichier texte. Les options sont `-l` pour line (nombre de ligne), `-w` pour word (nombre de mots) et `-m` pour le nombres de lettres. Il y a aussi l'option `-c` pour avoir la taille du fichier en bits. Pour l'utiliser, on fourni simplement en paramètre l'adresse du fichier texte :

```
Buzut$ wc -l test.rtf
      33 test.rtf
```

`wc` permet aussi facilement de savoir combien vous avez de fichiers/dossiers un dans répertoire donné, il suffit pour cela de rediriger la sortie d'un `ls` vers `wc` : `ls | wc` et le tour est joué !

`sort`

trier un fichier texte par ordre alphabétique. L'option `-r` permet d'effectuer un tri inverse, c'est à dire anti-alphabétique ou décroissant pour les nombres, et l'option `-R` permet un tri aléatoire, c'est le mode shuffle quoi ;) On n'oubliera pas non plus l'option `-u` qui permet d'éliminer les doublons. Enfin l'option `-o` permet de créer un nouveau fichier avec les résultats triés :

```
sort -o fichier_trie.txt fichier_en_bordel.txt
```

`uniq`

la commande `uniq` permet de dédoublonner un fichier. Il suffit de lui passer en paramètre l'adresse du fichier à dédoublonner et le nom du nouveau fichier à créer.

```
uniq doublons.txt no-doublons.txt
```

`cut`

couper dans un fichier texte. Pour couper toutes les lignes selon un nombre donnés de caractères, on utilisera l'option `-c`. `cut -c 2` conservera seulement les deux premiers caractères. On peut aussi donner un intervalle : `cut -c 2-4`, alors on

conservera uniquement les caractères deux à quatre. Exemple, « anticonstitutionnellement » sera transformé en « ntic ». Il est aussi possible de se servir de délimiteurs pour couper du texte, avec les options `-d` et `-f`. Les fichiers au formats .csv séparent les différents champs, les colonnes, par des point-virgules `;`. Dans un fichier où nous aurions trois champs, le nom, le prénom et la ville, si nous voulons extraire la ville, nous ferions comme ceci `cut -d ; -f 3` on indique le délimiteur après `-d` et le champ après `-f` (field veut dire champ en anglais).

`tar`

`tar` est l'utilitaire d'archivage. Il permet de regrouper des fichiers et des dossiers dans une seule archive. Les options intéressantes sont les suivantes : `tar -cvf` (create, verbose, file) permet de créer une archive, d'afficher tous les détails du processus (mode verbeux) et de tout mettre dans un dossier. Exemple :

```
tar -cvf nouvelle_archive.tar mon_dossier_a_archiver
```

Processus inverse, pour « détarrer » une archive, on utilise les options `-xvf` (eXtract, verbose, file) `tar -xvf archive.tar`. Les options `-tf` servent à afficher le contenu d'une archive sans l'ouvrir. Il est aussi possible de compresser et décompresser à la volée les archives `tar`, il faut rajouter pour cela l'option `-z` lors de la création ou l'ouverture de l'archive `tar -zcvf compress.tar.gz compress/`.

`gzip`

permet de compresser une archive tar au format zip. `gunzip archive.tar.gz`, il suffit ensuite d'utiliser la commande `gunzip` pour la dézipper.

`bzip2`

fonctionne exactement de la même manière que `gzip` mais compresse au format `bzip`. Pour décompresser l'archive, l'équivalent de `gunzip` est ici `bunzip2`.

`zcat` **`z...`**

`zcat`, `zmore` et `zless` remplissent les mêmes fonctions que `cat`, `more` et `less` mais à appliquer aux fichiers compressés.

`iconv`

permet de changer l'encodage d'un fichier. option `-f` pour préciser l'encodage d'origine et option `-t` pour celui de destination. Par défaut, `iconv` renvoit tout sur la sortie standard, donc si vous voulez directement envoyer les résultats dans un

fichier, il suffit de faire une petite redirection :

```
iconv -f UTF-8 -t UTF-17 fichier.txt >> new-encodage.txt
```

`wget`
: copie un fichier distant sur l'ordinateur. `wget http://www.site.org/rep/01/fichier.txt`.

`scp`
: vise à remplir la même fonction que la commande de copie `cp`, mais elle permet de copier les fichiers de manière sécurisé à travers le réseau; c'est à dire entre hôtes distants. De même qu'avec `cp`, l'option `-r` permet de copier un répertoire entier. `scp ficher_exemple login@ip_ou_adresse:adresse_de_destination`.

```
scp test.txt buzut@192.168.0.128:~/transfert

#et pour récupérer un fichier d'un hôte distant
scp buzut@192.168.0.128:~/movie.mkv ~/
```

`rsync`
: c'est un utilitaire qui permet de synchroniser entre eux des dossiers. Très pratique donc pour la sauvegarde. C'est pour ma part les options `-arv` que j'utilise. `-a` conserve les droits etc, `-r` permet la récursivité et `-v` pour le mode verbeux. Petit exemple de sauvegarde de mes photos de vacances :

```
rsync -arv photo backup_photo
```

Il est utile de préciser que si vous supprimez des fichiers dans le dossier source, `rsync` ne répercute pas la suppression dans le dossier de sauvegarde si vous ne lui adjoignez pas l'option `--delete`. Au cas où vous ne désiriez pas supprimer totalement les fichiers, il est possible de les placer dans un dossier séparé, options : `--backup --backup-dir=` Petit exemple pour la forme ?

```
rsync -arv photo ~/backup_photo --delete --backup --backup-dir=~/ba
```

Bien entendu, il est possible de faire une sauvegarde distante.

```
rsync -arv photo buzut@monserver:~/backup_photo --delete
```

Dernière chose, au cas où votre serveur n'écoute pas sur le port 22 en ssh, option -e, exemple pour un ssh sur le port 443 `-e 'ssh -p 443'`.

`file`

détermine le type d'un fichier indépendamment de son extension. Il suffit de lui fournir en paramètre le fichier à évaluer.

`split`

coupe un fichier en fichiers plus petits (-l préciser un nombre de lignes, -b préciser une taille en bytes (faites suivre la taille de K, M, G, T pour définir une unité différente). Pour créer des fichiers de 300 lignes `split -l 300 test.txt`, ou des fichiers de 1MB `split -b 1000000 mon_fichier`.

`sed`

dans les commandes qui permettent de manipuler du texte, `sed` est sans conteste l'une des plus puissantes. Remplacement selon une expression régulière, effacement de certaines expression où ligne selon un mot-clef donné… Seul inconvénient, qui dit puissance dit aussi complexité. `sed` ne s'explique pas en deux lignes et c'est pour cela que j'y ai consacré un article entier. Ça vaut le coup !

`awk`

`awk` est un langage de programmation à elle seule. Cette commande permet la recherche de chaines et l'exécution d'actions en fonction des motifs trouvés. D'une puissance redoutable, elle est aussi assez complexe. Je vous redirige donc vers des tutos en français ici ou là.

`locate`

cette commande permet de localiser un fichier sur le disque dur. `locate monfichier.txt`. La commande `locate` est très rapide car elle retrouve le fichier en consultant une base de données. Elle ne parcourt pas directement le disque dur à la recherche du fichier en question. L'inconvénient de ce procédé est que si le fichier est tout récent, il risque de ne pas encore être indexé, et `locate` ne vous sera alors d'aucun secours. On peut forcer la mise à jour de la base de données avec la commande `sudo updatedb`. On peut aussi se tourner vers la commande `find`.

`grep`

permet d'effectuer des recherches par expressions régulières. Dans sa forme la plus simple, `grep` permettra d'afficher la ligne contenant un mot clef (avec l'option `-o` on affiche seulement l'expression matchée), ceci depuis un fichier ou une autre

commande. Par exemple, si on veut afficher tous les processus ssh, on filtrera la commande `ps aux` avec grep : `ps aux | grep ssh`. `grep` permet également de rechercher dans le contenu de fichiers. Un exemple tiré de mon article sur la désinfection d'un WordPress où l'on cherche tout fichier `.php` potentiellement infecté :

```
# option -l pour matcher seulement certains types de fichiers (ici
# option -r pour la récursivité (recherche dans les sous-dossiers)
# option -E pour la syntaxe REGEXP étendue)
grep --include "*.php" -rlE 'viagra|pharma|Tadacip|eval|base64_dec
```

`find`

la commande `find` est bien plus puissante que `locate`, mais elle est aussi bien plus lente car elle parcourt le disque au fur et à mesure de la recherche. Sa syntaxe est la suivante `find /adresse_du_repertoire_de_recherche/ element_a_trouver`. Cette syntaxe n'est qu'une base. `find` permet en effet de rechercher selon une taille ou une date de dernier accès, mais encore d'effectuer des actions sur les fichiers trouvés, d'appeler une commande etc. Une page de man à lire donc…

Connaître la date de dernier accès est très pratique et puissant, quelques exemples :

```
# le plus simple chercher un fichier dont on connait le nom
# trouver tous les .htaccess dans /var/www
find /var/www -name ".htaccess"

# liste les fichiers dans www par date de dernière modification
find /www -type f -printf '%TY-%Tm-%Td %TT %p\n' | sort -r

# on peut être plus précis en affichant les fichiers modifiés dans
find /target_directory -type f -mmin -50

# ou les 24 dernières heures
find /target_directory -type f -mtime -24

# on peut enfin choisir une intervalle (entre hier et avant-hier)
find /target_directory -type f -mtime -48 ! -mtime -24

# last but not least, déplacer les fichiers qui ont plus de 24h (o
find /target_directory -type f -mtime -24
```

Vous noterez très certainement que nous utilisons ici `mtime` pour la dernière **m**odification. Il y a également :

- `mmin` pour les minutes,
- `atime` et `amin` pour le dernier **a**ccès,
- `ctime` et `cmin` pour le dernier **c**hangement.

Vous pouvez lire cet article et ce post pour bien comprendre les différences et implications *change time* et *modification time*.

Flux de redirection

`>`
: renvoyer le resultat dans un fichier (si celui-ci existe, il sera écrasé).

`>>`
: renvoyer le resultat dans un fichier (si celui ci existe déjà, ajoute le résultat à la fin) il existe deux sortie : 1 la sortie normale, 2 la sortie d’erreurs.

`2>`
: créer un fichier pour les erreurs. `2>>` existe aussi.

`2>&1`
: fusionne les deux sorties dans un seul fichier ex : `./script_de_la_mort.sh > fichier.log 2>&1`.

`<`
: prendre un fichier en entrée. ex : `cat < fichier.txt`.

`<<`
: prendre en entrée le clavier au fur et à mesure. Ceci nous permet de passer des données directement à une commande sans avoir besoin de créer de fichier. Mettons que nous voulions trier des prénoms par ordre alphabétique. Nous allons pour cela invoquer la commande `sort`, mais au lieu de créer un fichier texte puis de le faire réorganiser par `sort`, nous allons directement lui soumettre les noms en les entrant au clavier :

```
#on place un mot clef après le "<<" qui sert à délimiter les donné
#Ce mot clef est tout à fait arbitraire
#on appuie sur "entrer" après chaque prénom pour séparer les diffé
sort << STOP
> antoine
> clement
> quentin
> jordan
> mathilde
> clementine
> elena
> helena
> pierre
> STOP
antoine
clement
clementine
elena
helena
jordan
mathilde
pierre
quentin
```

`|`

rediriger le résultat d'une commande dans une autre. ex: `sort prenoms.txt | uniq`.

Multitâche et programme en arriere plan

`&`

mettre & à la fin d'une commande permet d'en lancer une autre sans attendre la fin de la première ex : `cp video.avi /users/buzut/desktop/copie-video.avi &`. Cette instruction permet aussi de passer la commande en arrière plan et redonne immédiatement accès au shell.

`nohup`
: lance le programme et le maintient même une fois la console fermée. Les sorties 1&2 sont redirigées vers `nohup.out`. Exemple :

```
nohup ffmpeg -i video-source.mkv -vcodec libx264 -preset slow vide
```

`ctrl` + `z`
: mettre en pause le process en cours.

`bg`
: passer le processus qui est en pause en arriere plan.

`fg`
: reprendre un process en premier plan (si plusieurs tournent en même temps, `fg %n°`).

`at`
: programme une tache à exécuter à une heure ultérieure ex : `at 18:22` ou `at now + 5hours` puis `ctlr` + `d`.

`atq`
: lister les jobs en attente.

`atrm`
: supprimer des jobs.

`sleep`
: cette commande permet de faire une pause entre l'exécution de deux commandes.
Exemple : `touch gt.txt && sleep 10 && rm gt.txt`
La pause est exprimée en seconde par défaut, il est cependant possible de changer cela en faisant suivre le nombre d'une unité : `m`, `h`, ou `d` pour respectivement les minutes, heures et jours.

`crontab`
: crontab est en fait une commande qui permet de lire et de modifier un fichier appelé la « crontab ».

`e`

modifier la crontab.

`l`
: afficher la crontab actuelle.

`r`
: supprimer votre crontab. Attention, la suppression est immédiate et sans confirmation !

`screen`
: multiplexeur de terminaux. Sous ce terme un peu barbare se cache en fait une sorte de terminal virtuel. Vous êtes au boulot, vous ouvrez un terminal et vous le nommez, vous lancez une tache (un script qui va réencoder plusieurs GB de vidéos par ex, ce qui prend du temps), et vous vous déconnectez, vous arrivez chez vous, vous réouvrez votre terminal et retrouvez votre tâche comme si tout était resté ouvert en face de vous.

```
# créer un screen
screen -S nom_du_screen

# pour le détacher (cela veut dire qu'il n'est plus affiché mais re
# il faut faire ctrl a puis d

# pour se rattacher à un screen détaché,
screen -r nom_du_screen

# pour lister les screens ouverts
screen -ls

# pour quitter/fermer un screen, comme pour fermer un terminal -->

# il n'est pas possible de s'attacher à un screen non détaché
# (screen non détaché dans un terminal auquel on tente de s'attache
# on peut en revanche partager un screen,
# on voit alors toutes les commandes tapées dans l'un ou l'autre de
# pour se connecter à un screen non détaché
screen -x nom_du_screen
```

Vous voudrez parfois remonter dans l'écran du screen, pour cela, il faut passer en *copy mode* avec les touches `ctrl` + `a` puis `[`. Vous pourrez alors naviguer avec les flèches

directionnelles. Enfin pour sortir de ce mode `ctrl` + `a` puis `esc`.

Droits, groupes et utilisateurs

`sudo`
exécuter une commande en tant que root.

`sudo su`
passer root et le rester.

`chmod`
changer les droits sur un fichier un ou dossier (option `-R` pour la récursivité dans tous les fichier et sous-dossier du dossier sur lequel on l'applique).

`adduser`
ajouter un utilisateur.

`passwd`
changer le mot de passe d'un user | ex : `passwd roger`.

`deluser`
supprimer un user (option `--remove-home` pour supprimer tous ses fichiers).

`addgroup`
créer un groupe.

`usermod`
modifie un utilisateur (options : `-l` pour changer le nom, `-g` pour lui assigner un groupe, `-G` pour lui assigner plusieurs groupes (séparés par des virgules), `-a` en complément de `-g` ou `-G`, ajouter des nouveaux groupes au lieu de tout redéfinir) ex, ajouter le groupe video à l'utilisateur buzut, sans supprimer les groupes auxquels il appartenait avant : `usermod -aG video buzut`.

`delgroup`
supprimer un groupe.

`groups`

vérifie dans quels groupes est un utilisateur `groups myuser`.

`chown`
change le propriétaire d'un fichier/dossier (ne peut s'utiliser qu'en root) option -R pour la récursivité.

`chgrp`
change le groupe propriétaire d'un fichier (équivalent à `chown user:group`).

`passwd`
Bien qu'il ne s'agisse pas, techniquement, d'une commande, il me semble important de connaître la structure du fichier `/etc/passwd`. Lequel regroupe l'ensemble des utilisateurs du système et de leurs informations.

```
# exemple pour l'utilisateur sensu
# 1  :2: 3 : 4 :            5              :     6    :    7
sensu:x:999:999:Sensu Monitoring Framework:/opt/sensu:/bin/false
```

1. Nom de l'utilisateur,
2. mot de passe (x signifie que le mdp est chiffré dans le fichier `/etc/shadow`,
3. l'id de l'utilisateur (0 est pour root et les id de 1 à 99 sont réservés pour les comptes prédéfinis),
4. l'id du groupe tel que défini dans `/etc/group`,
5. champ de commentaire,
6. répertoire « home » de l'utilisateur,
7. le shell par défaut (`/bin/false` et `/usr/sbin/nologin` signifient que l'utilisateur n'a pas de shell).

Système

`w`
qui est connecté et fait quoi.

`who`
qui est connecté.

`date`

donne l’heure.

`ntpdate`
synchronise l’heure avec un serveur ntp. Il faut préciser le serveur à la commande `ntpdate pool.ntp.org`. Par ailleurs, pensez que le port NTP (123 en UDP) doit être ouvert sur votre machine.

`uptime`
temps depuis mise en route + charge (charge moyenne 1 – 5 – 15 mn).

`free`
indique l’allocation de la ram et la mémoire libre restante.

`vmstat`
info ram, swap, cpu.

`proc/me…`
Le fichier `/proc/meminfo` contient de nombreuses informations sur la mémoire. Il suffit d’en afficher la sortir avec `cat /proc/meminfo`.

`tload`
affiche la charge CPU sous forme de graphique.

`ps -ef`
afficher tous les processus lancés. Alternativement, on peut utiliser la syntaxe BSD : `ps aux`.

`ps -ejH`
afficher process en arbre.

`ps -u`
lister les process lancés par un utilisateur donné ex: `ps -u buzut`.

`top`
l’activité du système en temps réel : load, RAM, SWAP processus… `top` a l’avantage d’être installé presque partout.

`htop`
c’est une version améliorée de top, un peu plus graphiques, les infos y sont plus

claires et il est possible de trier/ordonner l'affichage selon certains critères.

`glances`
: similaire à `top` et `htop`, `glances` est le tableau de bord de votre machine car il réuni en un coup d'œil toutes les métriques importantes : cpu, load, ram, swap, i/o disques, remplissage des disque. C'est l'œuvre de l'ami Nicolargo et son outil est maintenant intégré au dépôts des dernières versions Debian & co (je ne sais pas pour les autres distribs).

`atop`
: on a parlé ci-dessus de `top` et `htop`, mais il y a aussi `atop` qui est très utile lorsqu'on doit faire un diagnostique un peu plus poussé. De manière générale, j'utilise `htop` en remplacement de top car il est plus ergonomique et lisible, mais quand quelque chose cloche sur le serveur, direction `atop` ! Pour aborder sereinement la bête, je vous conseille la lecture de cet article de linuxpedia, admirable de clarté.

`iotop`
: dans la ligné des `*top`, voici `iotop` qui, comme son nom le laisse entendre, permet d'avoir un aperçu temps réel de l'I/O disque.

`swapoff`
: `swapoff -a` permet de désactiver le(s) swap tandis que son pendant `swapon -a` permet d'activer le(s) swap. L'exécution de `swapoff -a && swapon -a` permet donc de forcer la purge du swap.

`kill`
: tuer un processus (va demander son PID).

`kill -9`
: force à quitter.

`killall`
: quitte toutes les occurrences d'un programme.

`reboot`
: Redémarrer le système d'exploitation.

`shutdown`

Programmer un redémarrage ou un arret.

`poweroff`
bien qu'assez similaire à shutdown dans la mesure où elle permet d'éteindre le système, poweroff permet aussi selon les arguments qui lui sont passés, de rebooter ou de changer de runlevel. Vous pouvez aussi consulter ce thread [en] qui explique les différences entre `poweroff` et `shutdown`.

`halt`
permet « l'arrêt » du système. Je mets *arrêt* entre guillemets car le système peut rester sous tension avec cette commande (selon les options passés et les paramètres par défaut du système). Regardez les différences entre shutdown et halt [en].

`last`
historique des connexions.

`df`
remplissage des disques (l'option `-h` permet d'obtenir les tailles en « human readable »).

`mount`
Permet de monter le prériphérique d'un système de fichier sous un répertoire local. Par exemple, pour monter la partition `/dev/sdb1` au point de montage `/home` : `mount /dev/sdb1 /home/`. Inversement, nous utilisons `umount` pour démonter le volume.

`fdisk`
Permet de gérer les partitions. Affiche la table des partitions si on utilise l'option `-l`. Vous pouvez jetez un œil à l'article sur le partitionnement Linux pour savoir comment et pourquoi partitionner votre machine.

`parted`
Semblable à `fdisk`, `parted` supporte les partition GPT. L'option `-l` permet d'afficher la table des partitions.

`fsck`
Cet utilitaire permet de vérifier et de réparer le système de fichier.

`dd`

L'outil permet d'effectuer des opérations sur les disques, notamment de les effacer, par exemple `dd if=/dev/zero of=/dev/sdc` ou de cloner un disque ; ex : `dd if=/dev/sda1 of=/dev/sdb1 bs=64K conv=noerror,sync`.

`ddrescue`

Lorsqu'on fait face à un disque dur endommagé, il vaut mieux tenter de le cloner avec `ddrescue` qu'avec `dd`. Le preier est en effet dédié à cet usage. Il peut effectuer une première passe pour récupérer le maximum de données en ignorant les secteurs endommagés, puis dans une seconde passe il tentera de récupéré les données endommagées. La doc d'Archlinux détaille bien le processus de recovery.

`smartctl`

`smartctl` permet d'afficher les informations smart d'un disque. `smartctl -a /dev/sda` affiche toutes les informations à propos de sda. La fiche Wikipedia à propos du SMART fournit de bonnes explications sur les différentes données et leur interprétation.

`/proc/m...`

Fichier qui contient les infos sur vos RAID logiciels. On peut afficher ces informations en faisant un `cat` : `cat /proc/mdstat`. Lisez l'article sur les différents niveaux de RAID (simulateur intégré) pour trouver le RAID qui vous correspond Vous pouvez également consulter ce wiki [en] pour vous orienter dans la jungle des infos délivrées par `mdadm`..

`mdadm`

Commande qui permet d'obtenir des infos sur les RAID soft et de les paramétrer (sortir un disque de l'array, en rajouter un, le reconstruire en cas de disque défaillant…). Par exemple `mdadm --detail /dev/md0` donnera tous les détails lié à un l'array. Notez que cette commande est assez proche de l'option `--examine`, mais laquelle s'applique à un disque constitutif d'un raid `--examine /dev/sd*` et non au volume raid. Cette commande étant très riche et servant de nombreuses fonctions, je vous laisse vous référer au man pour plus de détails. On peut aussi trouver de précieuses informations sur le linux raid wiki.

`lsof`

list open files, dresse la liste des fichiers ouverts. Comme le fait remarquer l'ami MagiCrazy dans les commentaires, cette commande peut s'avérer bien utile pour voir quel fichier bloque le démontage d'un filesystem par exemple.

`hostname`

Affiche le nom d'hôte de la machine conformément à ce qui est écrit dans le fichier `/etc/hostname`.

`uname`

Infos sur le système et le matos.

`lsb_rel...`

`lsb_release -a` donne toutes les infos sur la distrib.

`lshw`

Donne une liste détaillée de l'hardware système tels que la configuration ram, la version du firmware, la configuration de la carte mère... Avec l'option `-short` vous obtiendrez une sortie plus digeste. L'option `-c network` s'avère aussi bien utile pour connaître le nom d'une interface réseau encore non configurée avec le standard de nommage systemD [en].

`lsblk`

Liste tous les devices de type bloc (disque dur).

`lspci`

Liste tous les périphériques PCI.

`lsusb`

Liste tous les périphériques USB.

`/proc/v...`

Fichier qui contient des infos sur le noyau. On peut afficher son contenu avec `cat` : `cat /proc/version`.

`/proc/c...`

Fichier qui contient des infos sur le processeur. On peut afficher son contenu avec `cat` : `cat /proc/cupuinfo`.

`dmideco...`

lit les info du bios.

`dmesg`

affiche les messages du buffer du noyaux.

`apt-cac…`
: gestion des paquets. Deux options sont très utiles `apt-cache search nom_paquet`, permet de chercher un paquet, et `apt-cache show`, permet d'obtenir des détails sur un paquet.

`apt-get`
: gestion des paquets. Les commandes que l'on utilisera le plus sont `update` (MAJ des sources de paquets dispos), `upgrade` (mise à jour du système et autres softs), `install` (`apt-get intall truc-à-installer` pour installer un nouveau logiciel et ses dépendances), `purge` (permet de désinstaller un paquet de manière plus « propre » que remove car cela efface aussi les fichiers de configuration).

`apt`
: apparue assez récemment, `apt` est décrit dans son man comme le front-end utilisateurs pour un usage plus interactif d'autres outils spécialisés tels que `apt-get` ou `apt-cache`. Il offre dans l'ensemble les mêmes possibilités que `apt-get`. Je retiens une commande toute particulière : `apt list --upgradable` qui permet de lister les packets qui seront mis à jour si l'ont fait un `upgrade` (avec `apt` ou `apt-get`).

`aptitude`
: c'est un autre utilitaire de paquets. Plus récent qu'apt-get, il est installé en parallèle de celui-ci sur Ubuntu et Debian. Préférez-le à apt-get. Il s'utilise dans l'ensemble comme apt-get mais est plus performant.

`add-apt…`
: `add-apt-repository` permet d'ajouter des dépôts alternatifs aux dépôts officiels. C'est très utile car les dépôts officiels ont souvent du retard sur les versions de logiciels que sortent les développeurs et certains logiciels en sont même absents. Ainsi, en ajoutant par exemple les dépôts des développeurs, vous pouvez bénéficiez des dernières versions juste en vous servant de `apt-get` ou `aptitude`, sans avoir besoin de compiler !

Par exemple, FFMPEG avait été supprimé des dépôts officiels d'Ubuntu (réintégré en version 15.04) au profit de Libav, son fork. Pour profiter des dernières versions de FFMPEG sans avoir à compiler manuellement à chaque fois :

```
sudo add-apt-repository ppa:kirillshkrogalev/ffmpeg-next

# on met notre liste de packets à jour
sudo apt-get update

# si ffmpeg n'est pas installé ont l'installe
sudo apt-get install ffmpeg

# s'il est déjà installé,
# un upgrade se chargera de le mettre à jour
sudo apt-get upgrade
```

Dernière astuce, ce n'est pas une commande spécifique, mais une combinaison de commandes qui permettent de rechercher un dépôt sur le système :

```
grep ^ /etc/apt/sources.list /etc/apt/sources.list.d/* | grep nom_
```

`apt-key`

Cette commande va bien souvent de paire avec l'ajout de dépôts puisqu'elle permet de gérer les clefs cryptographiques en validant l'authenticité. On utilisera le plus souvent `apt-key adv` pour ajouter de nouvelles clefs, Ubuntu-fr en détaille très bien l'usage. `apt-key list` permet de lister toutes les clefs installées et `apt-key del` permet d'effacer une clef.

`apt-cac...`

Autre commande bien pratique, `apt-cache madison nom_du_packet` affiche les différents dépôts liés à un packet donné et les versions actuelle de chacun d'entre eux.

`dpkg`

info sur les paquets installés (options pour lister tous les paquets, désinstaller etc) ex : liste des paquets installés `dpkg --get-selections`. L'option `-l` fourni également une liste exhaustive et avec une petite description de chaque packet, ce qui peut s'avérer très pratique. On constate parfois qu'un grand nombre de paquets sont marqués pour être désinstallés avec le tag `deinstall`, pour tout enlever d'un coup :

```
dpkg --purge `dpkg --get-selections | grep deinstall | cut -f1`
```

update ...

Si votre partition `/boot` est indépendante et qu'elle n'est pas très grande, il est probable qu'après un certain temps, vous deviez faire un peu de ménage, sans quoi l'espace nécessaire à une **mise à jour du kernel** est insuffisant. On devra donc supprimer les anciens noyaux, pour ce faire, :

```
apt-get purge $(dpkg -l linux-{image,headers}-"[0-9]*" | awk '/ii/
```

Il ne vous reste alors plus que le kernel actuellement utilisé et vous pouvez maintenant effectuer votre mise à jour sans problème. Vous trouverez plus de détails sur cette commande sur ce thread askubuntu.

`service`

Cette commande permet de gérer les services. Lancer, arrêter, lister les services du système etc. Par exemple, pour relancer nginx après un changement dans le fichier de config, on fera `service nginx restart` (notez que `reload` suffit dans certains cas). La commande `service --status-all`, bien pratique, permet de lister tous les services disponibles sur le système.

Le paramètre `status` s'avère souvent fort utile puisqu'il donne des informations sur un service n particulier, notamment s'il est actif ou non. Exemple `service nginx status`.

On peut également activer ou désactiver le démarrage automatique des services au boot, pour cela, on doit troquer la commande `service` pour la commande native `systemctl`. Prenons apache2 pour exemple :

```
# activer/désactiver un service
systemctl enable apache2
systemctl disable apache2

# vérifier si un service est lancé au boot
systemctl is-enabled apache2
```

`make`

permet de compiler un programme dont on détient les sources. En général on fait tout d'abord `./configure` [à lancer avec `--help` pour voir les différentes options de compilation] (lance le script de configuration qui vérifie la présence de toutes les dépendances, et écrit le fichier makefile qui contient les ordres de compilation), `make`, et enfin `make install` (elle installe le logiciel).

`update-...`

`update-rc.d` permet de configurer le démarrage ou l’arrêt automatique de service au démarrage de la machine ou selon le runlevel. On donne en argument le nom du service et l’action (remove ou default pour l’ajout) `update-rc.d -f apache2 remove`; `-f` permet de forcer l’effacement du lien symbolique même si le nom existe encore. On peut aussi placer un script de démarrage dans répertoire `/etc/init.d` ou le renseigner le fichier `/etc/rc.local` (qui a lui-même un lien symbolique dans /etc/init.d).

`/etc/pa...`

Fichier qui contient les différents comptes utilisateurs de la machine (ce qui inclut les comptes utilisés par les logiciels ex : www-data pour Apache). On peut afficher les informations avec `cat` : `cat /etc/passwd`.

`cat /et...`

Fichier qui contient les groupes utilisateurs de la machine (ce qui inclut les groupes utilisés par les logiciels ex : www-data pour Apache). On peut afficher les informations avec `cat` : `cat /etc/group`.

`which`

localiser une commande ex :

```
which cat
/bin/cat
```

`whereis`

localiser un fichier binaire.

Web

Web veut tout dire et rien dire à la fois, surtout dans la mesure où on parle déjà d’un serveur. Cependant, cette section concerne tout particulièrement les commandes liées aux serveurs web. On pense évidemment immédiatement à Apache2 mais d’autres pourraient s’y rajouter.

`a2ensite`
: activer un `vhost` Apache : `a2ensite buzut`

`a2dissi…`
: désactiver un `vhost` Apache : `a2dissite buzut`

`a2enmod`
: activer un module Apache : `a2enmod rewrite`

`a2dismod`
: désactiver un module Apache : `a2dismod rewrite`

`a2enconf`
: Sur le même modèle que pour les vhost et les modules, `a2enconf` et `a2disconf` permettent d'activer ou de désactiver des configurations. Par exemple, lorsqu'on ne renseigne pas explicitement le fichier d'`access.log` d'un vhost, Apache log le tout dans un fichier dédié : `other_vhosts_access.log`. Dans le cas où l'on ne voudrait pas de logs d'accès par exemple, on peut désactiver cette configuration : `a2disconf other-vhosts-access-log`.

`apache2…`
: `apache2ctl` permet d'une part d'agir en tant que script init (ce qui n'a pas grand intérêt puisqu'on utilise en général la commande `service`). Et d'autre part de controller le processus Apache et de récupérer des infos sur ce dernier. `apache2ctl -S` est assez utile puisqu'elle permet de voir comment apache interprète nos Vhost, l'option `-M` liste les modules Apache activés. Enfin, l'option `-help` vous en dira plus sur les autres commandes disponibles !

`curl`
: `curl` permet, comme `wget`, de récupérer un fichier depuis une url, mais s'il mérite sa place ici, c'est qu'il permet bien plus. C'est en effet le couteau suisse du HTTP. On l'utilisera en effet pour tester des requêtes dans différents formats, analyser les `HEADERS` etc.

```
# requête get classique
curl https://buzut.fr

# afficher les headers
curl https://buzut.fr -D -

# faire une requête de type X (HEAD, POST, PUT, PATCH, DELETE…)
curl -X HEAD https://buzut.fr

# passer des paramètres au format form data
curl -X POST --data "email=moi@mail.com&passwd=azerty" https://mons

# même requête avec les paramètres en request payload, JSON. (on a
curl -H "Content-Type: application/json" -X POST --data '{"email":
```

Gestion réseau

`iftop`
: dans la même veine que top, iftop sert à surveiller toutes les connexions réseau. Attention, iftop nécessite les privilèges root pour être lancé. Si vous n'êtes pas root, pensez à le faire précéder de sudo.

`speedom…`
: un peu plus graphique que `iftop`, `speedometer` monitor le traffic de vos entrées/sorties, permet de surveiller la progression d'un téléchargement, de savoir combien de temps il faudra pour transférer tel fichier ou encore de connaître la vitesse d'écriture de votre système.

`exim4`
: Exim est un MTA qui permet d'envoyer des email depuis le serveur. Sans lui (ou un autre MTA) la fonction `mail()` de php ou d'autres langages ne sera pas effective. Il en existe d'autres mais celui-ci est robuste, sécurisé, modulable et demande peu de ressources. Son installation est très simple, je l'explique dans cet article sur le logging.

`ifconfig`
: cette commande des plus indispensables permet d'obtenir des infos et de configurer les interfaces réseau. Employée sans argument, elle fournit des infos sur les

interfaces réseaux. Mais elle permet aussi de modifier la configuration. Par exemple pour changer une adresse mac, on utilisera la commande `ifconfig $INTERFACE ether $MAC`. Donc pour changer l'adresse mac de la carte ethernet, ce sera en général `ifconfig eth0 ether 5E:FF:56:A2:AF:15`. Notez cependant que cette commande, bien que toujours fonctionnelle, est dépréciée en faveur de la commande `ip` ci-dessous.

`ip`

cette commande permet d'afficher et de manipuler le routage et les interfaces. On s'en sert souvent pour lier ou supprimer une ip à une interface :

```
# lister toutes les addresses
ip addr

# ne lister que les informations ipv6
ip -6 addr

# lister une interface en particulier
ip addr show dev em2

# ajouter une adresse ipv4
ip addr add 192.168.0.7 dev eth0

# supprimer une ipv4
ip addr del 192.168.0.7 dev eth0

# pour la v6, il faut utiliser l'argument -6
ip -6 addr del 2574:104::b08b:c107 dev eth0
```

`ping`

permet de pinguer un client pour voir s'il est en ligne ou s'il répond au ping, ex `ping google.fr`. L'option `-c` permet de préciser le nombre de ping à envoyer avant que la commande ne s'arrête (elle prend donc en argument un nombre entier exemple : `ping -c 8 google.fr`), l'option `-f` permet de flooder, c'est à dire que la carte réseau enverra autant de ping qu'elle est capable d'envoyer par seconde.

`tracero...`

trace la route d'un paquet, routeur par routeur, jusqu'à sa destination. On peut utiliser indiféremment une ip ou un nom de domaine.

```
traceroute to google.com (173.194.67.113), 64 hops max, 52 byte pa
 1  fast3504 (192.168.1.254)  12.108 ms  1.239 ms  1.217 ms
 2  sl869-h01-31-38-37-254.dsl.sta.abo.bbox.fr (42.58.37.254)  33.
 3  173.la63.bsr01-lyo.net.bbox.fr (194.158.110.189)  32.684 ms  3
 4  be19.cbr01-cro.net.bbox.fr (212.194.171.16)  41.761 ms  41.636
 5  be1.cbr01-ntr.net.bbox.fr (212.194.171.1)  45.495 ms  42.217 m
 6  * * *
```

`dig ndd`

vérifier une correspondance dns. On peut directement spécifier l'ip du serveur dns à interroger, par exemple, `dig @8.8.8.8 buzut.fr`.

```
Buzut:~ Buzut$ dig @8.8.8.8 buzut.fr

; <<>> DiG 9.6-ESV-R4-P3 <<>> @8.8.8.8 buzut.fr
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2660
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL:

;; QUESTION SECTION:
;buzut.fr.                      IN      A

;; ANSWER SECTION:
buzut.fr.               43200   IN      A       213.186.33.4

;; Query time: 155 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Fri Jul 27 09:01:06 2012
;; MSG SIZE  rcvd: 42
```

`host`

permet de vérifier le reverse. Vous entrez donc une adresse ip et la commande vous retourne le nom de domaine associé.

`nslookup`

un peu dans le même genre que dig, `nslookup` fourni des infos sur le nom de domaine passé en argument, adresse ip, type de réponse DSN… Exemple : `nslookup`

`buzut.fr`.

```
Buzut:~ Buzut$ nslookup buzut.fr
Server:         212.27.40.240
Address:        212.27.40.240#53

Non-authoritative answer:
Name:   buzut.fr
Address: 213.186.33.4
```

`route`

cette commande d'une praticité sans pareil vous permet de visionner les routes, mais aussi, en lui spécifiant quelques arguments, de modifier les routes. Pour ajouter une route par défaut : `route add default addr ip` et pour supprimer une route par défaut : `route delete default`.

`arp -an`

cette commande permet de voir la table arp actuelle. C'est à dire la correspondance entre les adresses ip et mac sur votre réseau.

`nmap`

`nmap` est un outil qui scanne le réseau et les ports réseau d'une machine afin de voir lesquels sont ouverts et de détecter d'éventuelles failles sur les machines. Il permet aussi de détecter des machines connectées au réseau et bien plus encore. J'ai fait un petit billet sur nmap, qui reste un outil de référence dans le monde de la sécurité informatique.

Netstat

Ça c'est un morceau, et en faire le tour prendrait (au moins) un article entier. Quoi qu'il en soit cette commande est très pratique pour avoir un aperçu de ce qui se passe sur le réseau. On va donc voir directement quelques combinaisons de paramètres qui reviennent souvent.

`-nr`

pour la table de routage (revient au même que route).

`-i`

donne des statistiques sur les différentes interfaces réseau.

`-s`

personnellement je ne m'en sert que très rarement, mais c'est un résumé de toutes les stats réseaux, alors ça peut être utile de temps à autre.

`-uta`

liste toutes les connexions ouvertes (`u` pour udp `t` pour tcp et `a` pour afficher toutes les connexions (all)), à vous de jouer avec les paramètres pour filtrer un peu tout ça !

`-nap`

alternative à `-uta` (plus facile à retenir pour sa signification anglosaxone), cette commande donne un résultat que je trouve plus lisible. Elle affiche toutes les connexions (`a`) avec les adresses en format numérique (`n`) et donne le PID correspondant (`p`).

`-peanuts`

Dans le genre facile à retenir, on a là la combinaison des bons arguments ! Elle donnera un résultat un peu plus détaillé que `-nap` puisqu'elle ajoute UDP et TCP avec du détail (`-e` pour extended) et des statistiques (`-s`) pour toutes les connexions.

`-ltupn`

encore une variante qui nous permet de visualiser tous les services qui sont à l'écoute sur des ports (donc toutes les connexions ouvertes). Toutes les connexions tcp `-t` et udp `-u`, à l'écoute `-l` avec les programmes correspondant `-p` et accessoirement on demande de ne pas résoudre les nom de protocole avec `-n`.

`-l`

permet d'afficher toutes les connexion en écoute (listen), on aura donc `netstat -lt` pour toutes les conexions tcp en écoute.

`-n`

peut être pratique aussi car elle permet d'afficher les adresses en format numérique au lieu de tenter de déterminer le nom symbolique d'hôte, de port ou d'utilisateur.

`-p`

permet d'affiche le nom et le PID des processus propriétaires des connexions.

`netstat` est la commande historique pour lister les connexions. Cependant, elle a été dépréciée en faveur de `ss` qui est plus moderne et plus performante, surtout en présence de nombreuses connexions. La plupart des options sont les mêmes, vous ne serez donc pas perdu. Au besoin, un petit coup de `man ss` ne peut pas faire de mal.

Spécifique à OSX

On s'éloigne un peu de notre sujet d'origine, mais on est toujours dans les commandes utiles à connaitre dans un terminal. Les 3/4 du temps, les commandes linux fonctionnent sous mac, mais l'OS d'Apple possède quelques commandes qui lui sont propres. Certaines sont bien pratiques, je liste donc ici celle que j'ai déjà eu l'occasion d'utiliser.

`ipconfig`

`ipconfig getpacket interface` donne des infos sur la configuration ip, telles que l'adresse ip, l'ip des serveurs dns, le masque de sous domaine, le message dhcp… D'autres commandes existent avec `ipconfig`, mais `getpacket` est selon moi la plus intéressante, je ne parlerai donc ici pas des autres. Un petit exemple :

```
Buzut:~ Buzut$ ipconfig getpacket en1
op = BOOTREPLY
htype = 1
flags = 0
hlen = 6
hops = 0
xid = 3657777839
secs = 0
ciaddr = 0.0.0.0
yiaddr = 192.168.0.1
siaddr = 0.0.0.0
giaddr = 0.0.0.0
chaddr = 5E:FF:56:A2:AF:15
sname =
file =
options:
Options count is 7
```

```
dhcp_message_type (uint8): ACK 0x5
server_identifier (ip): 192.168.0.254
lease_time (uint32): 0xd2f00
subnet_mask (ip): 255.255.255.0
router (ip_mult): {192.168.0.254}
domain_name_server (ip_mult): {212.27.40.241, 212.27.40.240}
end (none):
```

`network…`

comme son nom l'indique, `networksetup` permet de configurer l'aspect réseau d'OSX. La commande est très très complète. On obtient par exemple la config DNS comme ceci : `networksetup getdnsservers ethernet | airport`. De la même manière, pour configurer les DNS, `networksetup setdnsservers ethernet | airport dns1 [dns2]`.

Déjà 52 réponses, rejoignez la discussion !

MagiCrazy dit :

13 septembre 2012 à 17 h 21 min

J'aurais ajouté « poweroff », qui a le pouvoir de permettre le wake on lan par défaut, alors que « halt » l'empêche. A vérifier quand même, ça fait longtemps =)

Y'a aussi « lsof », pratique pour savoir quel fichier empêche de désactiver un point de montage par exemple…

Répondre



Buzut dit :

14 septembre 2012 à 20 h 33 min

Merci de ces infos ! Je crois que poweroff permet bien de faire ça, en revanche je ne connaissais pas du tout lsof ! Je vais regarder et ajouter tout ça à la liste.

Répondre

 Buzut dit :

19 septembre 2012 à 16 h 30 min
Je ne suis pas arrivé à bien tirer au clair si poweroff permet le WoL tandis que shutdown ne le permet pas.
Le man reste muet là-dessus, si t'as une ressource à me passer, je suis preneur !
Répondre

 MagiCrazy dit :

19 septembre 2012 à 17 h 25 min
http://serverfault.com/a/191543

Il y a un semblant de réponse ici.
En gros, halt ça coupe le système mais pas l'alimentation, alors que poweroff envoie explicitement le signal à l'ACPI d'éteindre la machine.
Il est fort possible qu'aujourd'hui, sur nos machines, ce soit la même chose, je n'ai pas testé personnellement (pas encore ^^).
En tous les cas, j'utilise poweroff, parce que c'est moins chiant que halt ou shutdown à taper, avec les arguments qu'il faut…
Répondre

 Buzut dit :

19 septembre 2012 à 17 h 42 min
Yep je confirme pour halt ! J'avais même lancé un thread sur ubuntu FR (http://forum.ubuntu-fr.org/viewtopic.php?pid=9808131#p9808131) parce que halt ne m'éteignait plus le serveur électriquement parlant depuis la MAJ 12.04, tandis qu'avant ça avait le même effet qu'un poweroff. De plus d'après ton lien, les trois manières sont aujourd'hui à peu près équivalente… ¡ WTF !

Quoi qu'il en soit la machine doit resté continuer à tourner niveau hardware pour que le WoL fonctionne ? Faudrait faire les tests, ça me parait bizarre quand même !
Répondre

 MagiCrazy dit :

19 septembre 2012 à 19 h 20 min

> > > > Haha ! Ca se dénoue ! =)
> > > >
> > > > Je crois bien que tout ce qui reste allumé pour le WoL c'est juste la Carte ethernet, puisqu'il y a une configuration dans le BIOS pour gérer tout ça !


Buzut dit :

21 septembre 2012 à 15 h 46 min

Yep, c'est bien ça normalement. Mais avec halt, mes disques durs et ventilos (au moins ça, c'est vite remarqué…) continuent de tourner.

Et d'ailleurs, poweroff ne fait pas appel à halt justement ? Ahh !! ça m'énerve de ne pas trouver maintenant :evil:

Répondre


qwerty dit :

22 septembre 2012 à 8 h 27 min

Je pensais 101 comme 3 en binaire -_-'. Je suis grave. Merci du partage !

Répondre

> MagiCrazy dit :
>
> 22 septembre 2012 à 10 h 33 min
>
> 101 en binaire, ça fait 5 surtout =)
>
> Répondre
>
> > 
> > qwerty dit :
> >
> > 24 septembre 2012 à 20 h 15 min
> >
> > Mea Culpa !
> >
> > Répondre


Girafe dit :

5 juin 2013 à 0 h 20 min

Je cherche a afficher a partir d'un repertoire le contenu d'un numero de ligne (12 par

exemple). J'explique encore: j'ai un repertoire qui a 50 fichiers et pour afficher la ligne 12 (Exemple: elle contient « Station: 09 »)de chaque fichier, je dois avoir 50 lignes ayant:
Station: 09
Station: 10
Station: 12
Station: 14
....
Station: n
Répondre



Magic.Crazy dit :

5 juin 2013 à 1 h 14 min
C'est pas clair, mais je sens que AWK peut t'aider !

http://www.gnu.org/software/gawk/manual/gawk.html
Répondre



dvl dit :

22 mai 2014 à 20 h 30 min
Bonjour,

Sous Ubuntu, comment rediriger la communication d'un périphérique branché en USB mais dont les logiciels qui le font fonctionner ne communiquent que via le port COM ? Il faudrait donc que Linux fasse la transition COM 1 -> USB0 pour que le périph' fonctionne. J'ai testé quelques trouvailles mais sans succès.

C'est la seule chose qui me force encore à garder du windows dans un coin.
Répondre



Buzut dit :

24 mai 2014 à 10 h 53 min
Pour le coup, je n'en ai vraiment aucune idée !!

Mais je serai curieux de savoir si tu trouves une technique qui marche :)
Répondre


dvl dit :

7 novembre 2014 à 22 h 09 min

Finalement j'ai fait au plus simple, j'ai acheté le même périphérique modèle wifi et ça marche sans soucis, plus besoin d'USB et COM1. Et donc bye bye la partition winchose.

Répondre

 Buzut dit :

8 novembre 2014 à 12 h 59 min

Ça a le mérite de fonctionner ! Le problème quand on commence à bidouiller pour faire fonctionner des trucs classiques sur un ordi de tous les jours, en général c'est pas des solutions pérennes… Au moins là t'es tranquille

Répondre

patrick L dit :

12 septembre 2016 à 14 h 46 min

ln -s /dev/usb/hiddev0 /dev/com1

mais je pense pas que ca puisse se faire… l'usb envoie chaque message vers TOUS les periphériques en etoile.

en ligne série on envoie en direct on a pas forcement de numero de périphérique à envoyer. Pour une imprimante serie je me moquais d'envoyer un numero. D'accord ensuite je pouvais faire du hard de facon à envoyer sur la ligne en mettant un numéro de code.. tout la ligne recevait le message mais pour emettre. je devais mettre un signal rts de demande d'emettre et vérifier le cts qui autorise.

le usb marche pas du tout pareil. on a systematiquement un numero de périphérique qui est géré par une administration.

si ton imprimante est en usb il faut que tu ais le numero de usb… les deux vendor_id et usb_id

sur un terminal tapes lsusb et il faut que tu aies le protocole pour envoyer sur usb.

Répondre


gui dit :

18 juin 2014 à 23 h 32 min

Bonsoir,

je découvre les commandes linux.
Possédant un Nas, je souhaiterais vérifier le transfert de fichiers.
Est-il possible de visualiser si des transferts de fichiers sont en cours.
Que ce soit via le réseau ou d'un Usb vers le Nas.
Merci

Répondre

> 
> Buzut dit :
>
> 19 juin 2014 à 20 h 32 min
>
> Je te conseille de creuser du côté de speedometer pour ça. En revanche pour l'usb, je ne sais pas si ça fera l'affaire. Tu pourras déduire ça des lectures à partir du disque. Cependant, il doit y avoir des outils dédiés pour l'usb. Une petite recherche dans ton moteur favoris t'en dira plus !
>
> Répondre
>
> > 
> > Gui dit :
> >
> > 19 juin 2014 à 21 h 07 min
> >
> > Merci
> >
> > Répondre


Cascador dit :

6 juillet 2014 à 9 h 46 min

Hello,

Je pense qu'il serait de bon ton de rajouter les commandes awk et shutdown.

Les petites erreurs :
-e « ssh -p 443"
passer rot et le rester
apt-cahche search

elle fourni

Il faudrait préciser que ça écrase le fichier si il existe :
> : renvoyer le resultat dans un fichier

Excellent article !

Tcho !
Répondre



Buzut dit :

26 août 2014 à 12 h 58 min
Merci pour ton commentaire, tes suggestions et corrections, j'ai pris tout ça en compte.

À plus !
Répondre

philippe bdx dit :

21 septembre 2014 à 5 h 18 min
hellp :

Bizut; j'ai trouver ta page en cherchant un autre truc et pour le coup elle est dans mes favoris ;)

Une chose c'est quoi ton terminal, sur un mac ?

ciao merci

philipe midle class user « sous » ubnutu
Répondre



Buzut dit :

21 septembre 2014 à 9 h 38 min
Salut,

J'utilise tout simmplement Terminal.app, la version par défaut de Mac OSX

https://en.wikipedia.org/wiki/Terminal_%28OS_X%29

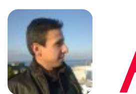
Très bon choix pour Ubuntu que j'utilise aussi dans sa version serveur.

À bientôt :)
Répondre


Rado dit :

3 octobre 2014 à 15 h 58 min
Bjr,

y-a-t-il une commande ou une combinaison de commande pour re-detecter/re-installer les périphériques (cartes réseaux, sons, ...)
Répondre


Ahmed dit :

10 mars 2015 à 17 h 05 min
C'est incroyable le nombre de commandes et combinaisons possibles qu'il est possible d 'utiliser sous LINUX! C'est vrai qu'il y en a tellement que ca devient difficile de tout retenir.. Hop imprimé ! :D
Répondre


Buzut dit :

11 mars 2015 à 9 h 24 min
Je suis content que ça puisse servir !
Répondre


Souki dit :

21 août 2015 à 11 h 13 min
j'ai besoin de la commande qui permet de visualiser les partages d'une machine stp..Heelp
Répondre



Busut dit :

21 août 2015 à 11 h 19 min

Les partages d'une machine?? C'est à dire?

Répondre

souki dit :

21 août 2015 à 11 h 24 min

pour avoir la liste des ressources partagées sur un réseau ?

Répondre

Buzut dit :

21 août 2015 à 11 h 35 min

netstat peut te fournir ça. Tu veux afficher toutes les connexions tcp `-t` et udp `-u`, à l'écoute `-l` avec les programmes correspondant `-p`. Tu peux même lui demander de ne pas résoudre les protocoles `-n`. Ce qui nous donne `netstat -ltupn`

Répondre

souki dit :

21 août 2015 à 11 h 40 min

Merciiiii bcp pour ta réponse :)

safaa dit :

25 octobre 2015 à 20 h 31 min

j'ai besoin de les commandes qui permet de faire une fiche technique de mon disque dur

Répondre

Frédo 31320 dit :

1 décembre 2015 à 10 h 35 min

Bonjour,

Très intéressant tout ça! :-)
J'ai toutefois une question…

Nos utilisateurs (qui n'ont pas les droits root sur les machines) et qui utilisent une session utilisateur dédiée peuvent s'ils le souhaitent modifier la configuration réseau via l'interface graphique. Pas bon du tout…. :-P
Le simple fait de cocher « Connexion système » au même endroit sur la session admin résout le problème…
Le hic est que le parc est assez conséquent (environ 400 machines) et qu'il n'est pas possible de faire cette modif poste à poste sur tout le parc sans avoir à recloner…
Je n'ai pas trouvé la commande qui permet cela et qui permettrait de scripter cette manip.
Quelqu'un a une idée? Merci! ;-)
Répondre

>  Buzut dit :
>
> 2 décembre 2015 à 9 h 24 min
> Tu peux peut être desactiver le network manager cf dans cet article
> http://www.sitepoint.com/ubuntu-12-04-lts-precise-pangolin-networking-tips-and-tricks/
> Répondre

 Alain dit :

9 décembre 2015 à 18 h 42 min
Bonsoir et merci pour cet article.
Je crée un fichier compressé sur un serveur Raspberry avec la commande
tar zvcf – ./today | ssh osmc@192.168.1.42 « cat > /media/raspdisk/abach/today.tar.gz »
Je voudrais que ce fichier today.tar.gz contienne exactement ce que j'ai dans ./today, que j'y ajoute ou supprime des fichiers.
Comme je ne sais pas s'il existe des options à la première partie de ma ligne de commande, je voudrais supprimer le fichier compressé avant de lancer cette commande.
J'ai cherché sur le Net mais là, je suis sec.
Avez-vous une solution ?
Répondre

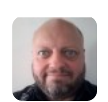

Buzut dit :

10 décembre 2015 à 9 h 32 min

Je nai rien qui me vienne a l'esorit en une ligne. Si jetais toi je ferai un petit script transfer_today.sh :
1 tu zippes
2 tu transferts avec scp
3 tu effaces le zip local

Et voila. Tu peux ensuite le mettre en cron et avoir la commande s'executer tous les jours.

Répondre


Alain dit :

10 décembre 2015 à 9 h 36 min

La réponse c'est faite d'elle même. Ma ligne de commande écrase systématiquement le fichier compressé sur le serveur distant.
C'est, certes, un peu plus long qu'une synchro mais ça fait le job

Répondre

Lapinu dit :

15 février 2016 à 10 h 45 min

Je vdrai savoir comment faire un programme qui Vérifie qui est connecté'

Répondre


max dit :

11 avril 2016 à 10 h 32 min

salut merci pour ses renseignements précieux pour noob comme moi ^^

Répondre


pacory dit :

4 mai 2016 à 22 h 01 min

erreur ntpdate synchronisation avec un serveur ntp et pas ftp
Répondre

> 
>
> 6 mai 2016 à 16 h 44 min
> C'est corrigé, merci !
> Répondre



24 octobre 2016 à 17 h 21 min
Bel effort de rédaction, bien présenté, facile à lire,…
Je me suis toujours dit que je ferais une page comme celle-ci et le manque de temps…
Enfin bref… Merci !
Répondre

> 
>
> 24 octobre 2016 à 17 h 31 min
> Merci pour ton commentaire ! Si d'aventures il te semble qu'une commande manque à l'appel, n'hésites pas à me le signaler :)
> Répondre



28 novembre 2016 à 16 h 08 min
bonjour

dans la liste des commandes, donc à partir d'un terminal,
je cherche la commande qui permet d'indexer un lecteur distant.

J'apporte une précision ici :
par distant, j'entends les NAS mais aussi les répertoires se trouvant par exemple sur le HDD d'une box ou un hdd en usb ou/et en ethernet – ce qui est de mon point de vue sensiblement équivalent à un nas.

Si l'indexation n'est pas possible sous ubuntu 16.04,

peut-être existe t'il un paquet ou un logiciel capable de faire cela ?

Par extension, si vous aviez une ligne de commande permettant de détecter les fichiers doublons …. je serai intéressé (ou s'il existe un logiciel …)

En tous cas merci de m'avoir lu

A buzut.fr
bon site, clair sans fioriture, cela change de CCM qui fatigue la vue

Cordialement
Répondre

> 
> Buzut dit :
>
> 28 novembre 2016 à 18 h 19 min
> Hello!
> Je n'ai jamais eu l'occasion de tester le montage de NFS. Il y a une doc plutôt complète sur redhat, je pense que c'est applicable sur de la Debian. Si tu testes ça, ton feedback sera le bienvenu.
>
> Pour ce qui est des doublons il faut regarder du côté de fslint qui est inclut dans les dépôts de la plupart des distrib.
> Répondre

elhadj_oumar_bah dit :

26 décembre 2016 à 0 h 05 min
en faite je besoin de l'aide je dois configurer et installer des services DNS et DHCP sous débian mais à vrai dire j'arrive pas à me retrouver.quelqu'un pour m'aider
Répondre

safwen dit :

22 janvier 2017 à 14 h 18 min
Bravo.
Une bonne collection.
Répondre


Buzut dit :

22 janvier 2017 à 18 h 17 min

Merci :)

Répondre


Franckito dit :

28 février 2017 à 11 h 33 min

Bonjour.
Cette page est très complète et bien utile pour les débutants. Je pense qu'il peut être utile de préciser trois petites choses :
– l'option -u de la commande sort est très utile pour supprimer les doublons pendant le tri. Cela évite de piper le résultat dans un uniq, ce qui consomme plus de ressources.
– dans le même esprit, l'option -o de grep, qui permet de n'afficher que la partie de la ligne qui correspond au motif recherché, et évite donc de piper la sortie dans un cut ou un awk.
– la commande touch a pour but de modifier l'horodatage d'un fichier. Si le fichier n'existe pas, il est effectivement créé mais ce n'est pas le but de la commande. Pour créer très vite un fichier on peut également juste saisir « > fichier », mais ce n'est pas le but des redirections, juste une utilisation particulière. Je sais, je chipote.

Répondre


Buzut dit :

28 février 2017 à 20 h 29 min

Merci pour ton commentaire et les précisions ! Je modifie tout ça ASAP.

Répondre

greg dit :

15 mars 2017 à 19 h 33 min

comment supprimer le mot de passe de son disque dur installé sous linux mint ou Lubuntu (je ne sais meme plus quelle version j'avais utilisé). Merci d'avance, je souhaite récupérer mon ssd bloqué.

Répondre

Laisser un commentaire

Votre adresse de messagerie ne sera pas publiée. Les champs obligatoires sont indiqués avec *

Que souhaitez-vous dire ? *

Commentaire

Quel est votre nom ? *

Nom

Quel est votre email ? *

Email

Vous avez un site web ? Partagez-le !

url

Notifiez-moi des commentaires à venir via e-mail. Vous pouvez aussi vous abonner sans commenter.

Notifications : Seulement les réponses à mes commentaires

Laisser un commentaire

Catégories

Linux
Serveur
JavaScript
HTML/CSS
PHP
WordPress
Marketing